



## **Service Industriel de l'Aéronautique**

AIA de Cuers-Pierrefeu

Rapport de stage de Valentin Mercenaro

Professeur référent : M. Sébastien Pioch

Maître de stage : M. Éric Rieuvernet

## Remerciement

Je tiens à remercier toutes les personnes qui m'ont aidé et accompagné à obtenir ce stage ainsi qu'à le mener à bien.

Je voudrais remercier dans l'entreprise qui m'a accueilli, l'AIA-CP :

- M. Éric RIEUVERNET mon tuteur de stage qui a accepté de me prendre qui a eu la patience de m'aider, corriger mes erreurs et répondre à nombre de mes questions.
- M. Alexis BEYNIX qui a également été présent pour m'aider en cas de besoin que cela soit en rapport avec mon projet ou non.
- M. Arnaud BALDUCCI, ingénieur travaillant sur la même plateforme que moi, pour sa bonne humeur et son humour.
- Mme. Audrey NURIT qui m'a aidé en l'absence de mon tuteur pour toute la partie accès et l'administration.
- Toute l'équipe du pôle conception pour leur accueil qui a été très agréable.

Je voudrais également remercier :

- M. Jean-Marc NATALI et Mme. Sandrine MACIGNO mes contacts à l'AIA sans qui je n'aurais pas pu obtenir ce stage incroyable.
- Mme. Arrivée pour ses enseignements, qui a eu la patience de me corriger à de nombreuses reprises lors de la rédaction de ma lettre de motivation et mon cv.

## Table des matières

Introduction .....	2
1. L'entreprise .....	3
1.1. Secteur d'activité.....	3
1.2. Spécificité .....	4
2. Le projet .....	6
2.1. Environnement de travail .....	6
2.2. Présentation du projet .....	7
2.2.1. Contexte .....	7
2.2.2. Objectif.....	8
2.2.3. Contraintes.....	9
2.3. Présentation du déroulé du suivi de projet .....	10
2.3.1. Activité .....	10
2.3.2. Mission secondaire .....	17
3. Bilan d'activité .....	17
3.1. Présentation des résultats .....	17
3.2. Prolongement de mission possible.....	18
3.3. Apports personnels .....	18
Conclusion.....	19
Glossaire et liste des figures .....	20
Annexes.....	22

## Introduction

Dans le cadre de ma formation de BUT 2 GEII j'ai réalisé mon stage en entreprise à l'AIA CP, les Ateliers Industriels de l'Aéronautique de Cuers-Pierrefeu pour une durée de 8 semaines. Durant ce stage j'ai été affecté au pôle conception où ma mission m'a été confiée. Comment mener à bien l'adaptation d'un simulateur d'entraînement de l'armée en une « attraction » grand public ?

J'ai choisi de réaliser mon stage au sein de cette entreprise car elle me permet de me rapprocher au plus de mon projet d'avenir actuelle, travailler en tant qu'ingénieur sur des drones militaires. Bien que la base ne s'occupe pas de drones elle s'occupe d'aéronefs et d'hélicoptères militaires. La division dans laquelle j'ai effectué mon stage me permettant d'avoir un bon aperçu de l'utilisation de l'informatique et de la programmation dans l'aéronautique.

Pour plus de compréhension dans ce rapport sera expliqué ce qu'est l'AIA CP, sa place sur le secteur de la maintenance aéronautique ainsi que ses spécificités. Ensuite sera développé la partie projet, le contexte du projet, l'état dans lequel il était à mon arrivée, le travail qui à réaliser dessus, la mission qui m'a été confiée et à quel niveau je devais aider dans ce projet, les problèmes et contraintes rencontrées puis enfin l'état du projet à l'issue de mon stage.

# 1. L'entreprise

## 1.1. Secteur d'activité

La France et sa population sont sous la protection du ministère des armées, ce ministère des armées est composé de 3 branches, l'Armée de Terre, l'Armée de l'Air et de l'Espace et la Marine Nationale.

L'armée de l'Air et de l'Espace est chargée de la sûreté de l'espace aérien Français et dans le monde selon les missions, mais est également chargée de la dissuasion nucléaire. Elle est représentée par des services interarmées, la direction de maintien aéronautique (DMAé) et la direction centrale du Service Industriel de l'Aéronautique (SIAé).

La SIAé est chargée du maintien en condition opérationnelle (MCO) des principaux aéronefs des forces armées françaises ainsi que de leurs matériels. Afin d'assurer cette fonction, la SIAé est répartie sur 5 sites sur le territoire Français, les AIA.

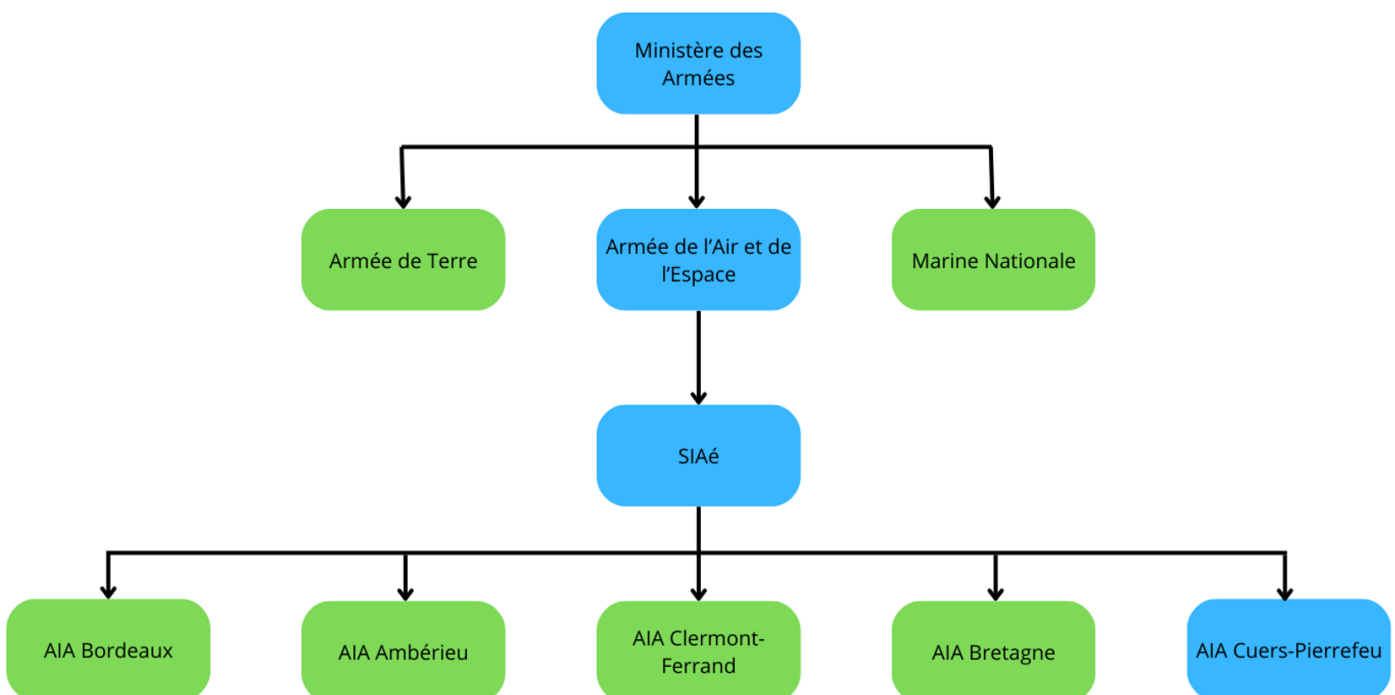


Figure 1. Hiérarchie des structures dont dépend l'AIA-CP

## 1.2. Spécificité

- AIA de Bordeaux (AIA) : spécialisé dans le MCO et l'expertise de moteurs d'aéronefs militaires. Ayant plus de 1000 employés ce site est notamment chargé de la maintenance des turboréacteurs du rafale, du mirage 2000 et de l'Alpha Jet, mais également des turbopropulseurs de l'Atlas A400M, du Hawkeye ainsi que des turbomoteurs d'hélicoptères (Lynx, Tigre, NH90).
- AIA d'Ambérieu (AIA AB) : ce site répartit ses compétences dans 4 domaines d'activité, la météorologie, les systèmes d'informations et de communication, la Sécurité-Sauvetage-Survie et tout ce qui est Matériels de Servitudes Embarqués et d'Environnement aéronautique.
- AIA de Clermont-Ferrand (AIA CF) : l'AIA de Clermont-Ferrand, employant 1300 personnes, est spécialisé dans le maintien en conditions opérationnelles d'aéronefs militaires tels que l'A400M Atlas, le C130H, l'Alpha Jet, le Tigre, le Puma, la Gazelle. Ils sont chargés de la maintenance de cellules d'aéronefs, la maintenance d'équipements d'aéronefs, la modification et la modernisation d'aéronefs essentiellement armée de l'air.
- AIA de Bretagne : comportant 675 employés répartie en 3 bases aéronavales, Landivisiau, Lanvéoc et Lann-Bihoué, ce site est chargé de la maintenance de niveau 2 des aéronefs et des simulateurs de la Marine Nationale. Ils interviennent sur le Rafale et ses moteurs M88, la maintenance de l'Atlantique 2 et équipements du Hawkeye ainsi que sur le NH90 et l'Alouette III.
- AIA de Cuers-Pierrefeu (AIA CP) : composé de 1000 employés, la principale mission de l'AIA CP est d'assurer la maintenance du Niveau de Soutien Industriel (NSI) des cellules et équipements des aéronefs essentiellement marines qui lui sont confiés comme l'Atlantique 2, le Hawkeye, le NH90 ou le dauphin/panther. Également équipé de deux niches technologiques, le pôle conception et la division radômes & composite, il assure le MCO de simulateurs, d'entraîneurs tactiques et de pilotage, de

bancs de tests et de systèmes opérationnels sol, la fabrication et le MCO de radômes d'aéronefs militaires (le nez des aéronefs).

C'est dans l'une de ces niches technologiques, le pôle conception, dans le service simulateurs, d'entraîneurs tactiques et de pilotage que j'ai réalisé mon stage.

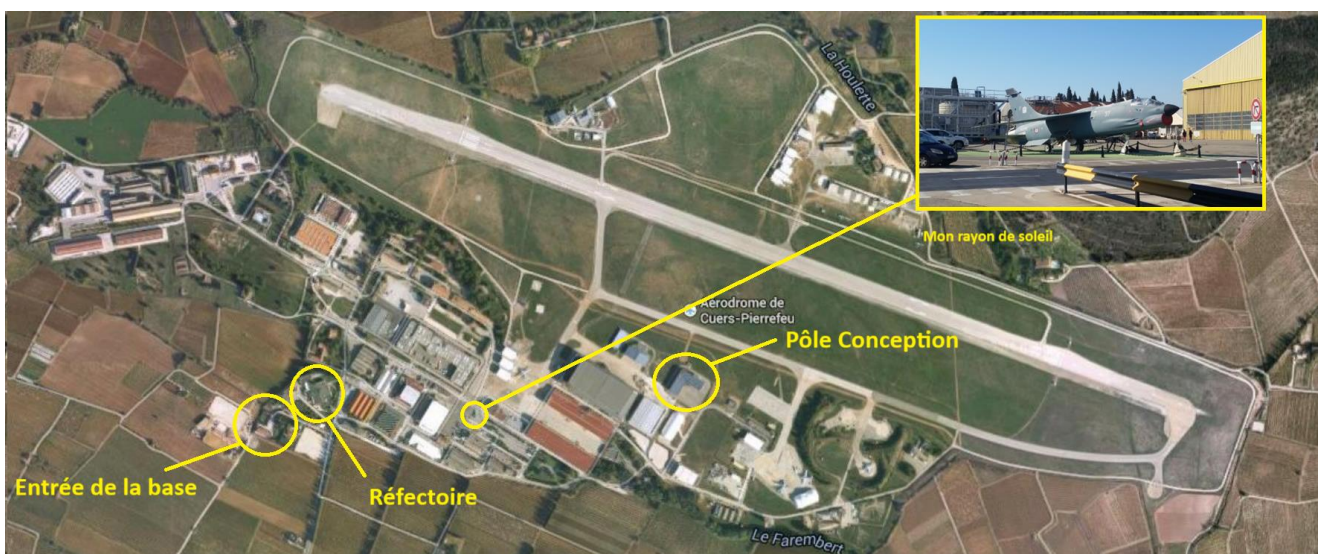


Figure 2. Vue aérienne de l'AIA-CP

La DGA envoie ses demandes de faisabilité suite aux besoins émis par les forces armées au service commercial du SIAé, cette demande est étudiée puis acceptée selon les moyens matériels et financiers à disposition. Si le projet est accepté le Pôle Conception entame la réalisation du projet. Il est également chargé du MCO et de la création de simulateurs, d'entraîneurs tactiques et de pilotages, de banc de test et de systèmes opérationnels sol ou embarqués. Dans le bâtiment du PC se trouve des plateformes systèmes représentant l'ATL2 et l'Hawkeye ainsi que des plateformes de simulation représentatif de l'entraîneur tactique panther et un simulateur de vol de Cirrus SR20/22.

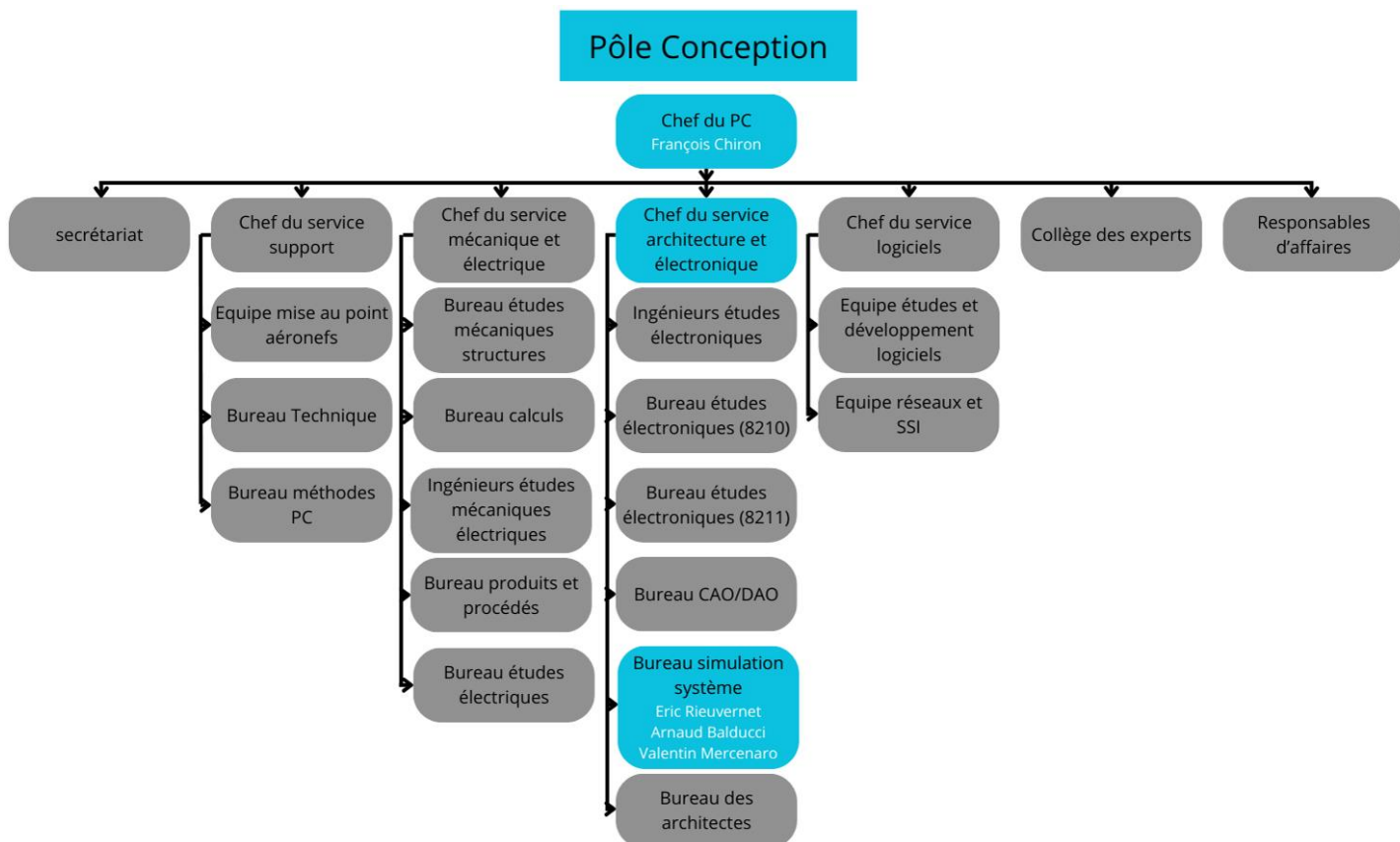


Figure 3. Organigramme du pôle conception

## 2. Le projet

### 2.1. Environnement de travail

Le bureau dans lequel j'ai travaillé est une plateforme, un espace fermé par une porte sécurisée accessible uniquement aux personnes habilitées. Cette plateforme est dédiée à la conception et à la maintenance des entraîneurs tactiques et vol. En plus du simulateur CIRRUS que j'ai étudié, on y trouve également l'entraîneur tactique d'un Dauphin Panther.

Pour effectuer la tâche qui m'était demandé j'avais à disposition 3 espaces différents, un poste de développement sur lequel était développé l'interface visuelle ainsi que le code (annexe 1).

Un post de test équipé du simulateur de vol X-plane, relié au poste de développement permettant de tester rapidement le programme en cours de développement. (Annexe 2)



Un poste comportant 3 pc reliés entre eux permettant de contrôler le vrai simulateur sur lequel seule la version finale fonctionnelle des programmes est implantée, un poste représentatif de la version finale du simulateur de vol (Annexe 3). Le simulateur avec X-plane implanté sur une machine à vérins avec un écran en face. Une fois qu'il a été vérifié que le programme fonctionne sur le simulateur il n'est plus touché. (Annexe 4)

La communication entre les différentes machines est décrite en annexe 5.

## 2.2. Présentation du projet

### 2.2.1. Contexte

Le premier contact qu'ont les apprentis pilotes avec l'aviation se fait sur un simulateur de SR20/22 qui a été développé au Pôle Conception de l'AIA-CP et qui est implanté à l'école militaire de Salon-de-Provence. Il permet ainsi d'apprendre les bases du pilotage aux élèves tout en étant capable de s'entraîner en cas de panne pour les instructeurs, les pannes sont simulées via le poste instructeur. Cela permet de réaliser des économies pour les entraînements tout en diminuant les risques, de cette façon lorsque les pilotes se retrouveront dans un vrai avion ils seront préparés en cas de problème. Pour cette raison le poste instructeur Cirrus est très complet, comporte beaucoup de pages avec énormément d'interaction possible, le simulateur étant presque entièrement paramétrable à partir du PI.

Le SIRPA a demandé le développement d'un simulateur ayant pour but de motiver les gens à devenir pilote au travers d'une attraction, suscité leur intérêt pour l'aviation au travers d'une expérience de vol réaliste. Un simulateur sur vérin qui serait présent lors d'exposition ou de portes ouvertes militaires afin de le faire essayer aux visiteurs.

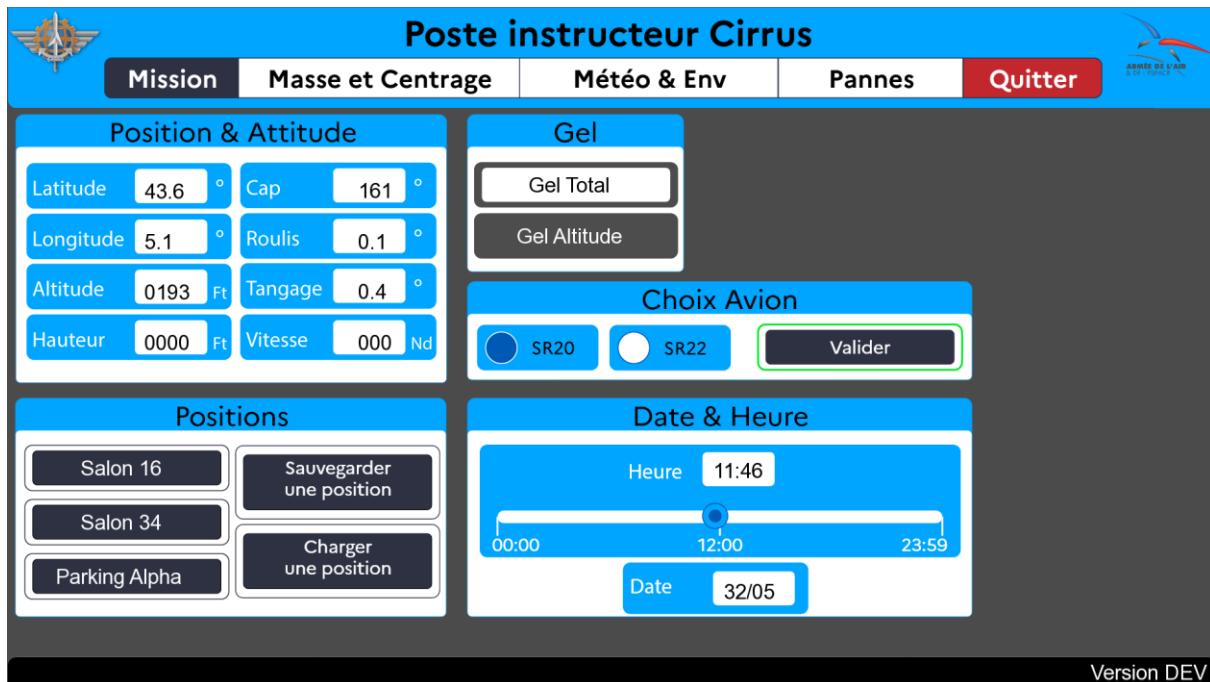


Figure 4. Poste instructeur Cirrus

### 2.2.2. Objectif

Le simulateur équipé du PI Cirrus est déjà en fonction depuis plusieurs années à la base de Salon-de-Provence, La mission qui m'a été confiée consistait à adapter le PI Cirrus en fonction de son utilisation prévue. Le PI SIRPA doit être simple d'utilisation, se limitant à une page et quelques boutons afin de pouvoir enchaîner les passages rapidement tout en offrant une expérience de vol réaliste dans des conditions idéales adaptées à des personnes inexpérimentées.

### 2.2.3. Contraintes

#### Programme :

Le langage de programmation utilisé dans le code est le C++, un langage déjà étudié en cours, cependant, le niveau de programmation du simulateur est bien plus avancé que celui abordé en classe, ce qui m'a donné l'opportunité d'en découvrir des aspects plus complexes. De plus, le code du PI Cirrus est un mélange de différentes méthodes de programmation, reflétant la manière de travailler des nombreuses personnes qui ont travaillé dessus avant moi.

#### Logiciel :

Visual studio 2010, une ancienne version de Visual Studio, le logiciel ne m'est pas inconnu sur des versions plus récentes.

La réalisation de l'aspect graphique de l'Interface Homme-Machine (IHM) a été réalisé grâce à un logiciel spécifique : VirtualInstrumentDesigner (développé par Agenium pour l'AIA). Ce logiciel permet de créer une interface, et de récupérer dans un programme toute interaction avec l'interface (appui sur un bouton par exemple).

La liaison entre le programme et la simulation était quant à elle réalisée par une bibliothèque spécifique elle aussi : X-plane Connect (développée par la NASA, en libre accès).

Photo filtre, un logiciel semblable à Paint, n'ayant pas de logiciel pro comme PhotoShope ou Adobe les boutons et cases ont été réalisé à l'aide de Photo filtre.

X-plane un logiciel de simulateur d'avion semblable à Flight Simulator.

## 2.3. Présentation du déroulé du suivi de projet

### 2.3.1. Activité

Afin de comprendre le fonctionnement du simulateur il m'a été possible d'essayer le simulateur présent dans la salle, puis analysé le code et l'IHM précédent pour comprendre ce qu'avaient fait les autres avant moi.

La représentation 3D du simulateur sur SolidWorks a également été étudiée. Étant donné que de nombreuses parties du simulateur sont imprimées en 3D, il est important de maîtriser SolidWorks afin de pouvoir recréer ou modifier une pièce en cas de maintenance.

La réalisation du poste instructeur SIRPA s'est réalisé en plusieurs étapes.

#### 1. Réfléchir à comment organiser mon PI

La seule consigne donnée était de créer un poste instructeur en tenant compte de son usage. La liberté totale a été accordée pour imaginer et proposer des idées, qui étaient ensuite validées ou non par mon tuteur. Le poste instructeur devait être à la fois minimaliste et très intuitif.

Toutes les idées ont été notées, certaines ont été retenues, d'autres non. Les idées conservées sont les suivantes :

À chaque sélection d'un point de départ essence, masse et centrage de la masse, heure, température, pression QNH (pression atmosphérique au niveau de la mer) doivent être réinitialisés afin que chaque personne ait la même expérience de vol. Des boutons définissant un point de départ (imposé Marseille Marignane et Salon de Provence), un bouton pause, un bouton d'arrêt et des boutons permettant de gérer la plateforme à vérins.

2. Reprendre l'IHM en entier afin d'en faire quelque chose de simple et compréhensible par n'importe qui. Pour cela, il a donc été nécessaire de créer les images au format PNG, puis de configurer l'IHM selon les spécifications souhaitées. Toutes les images utilisées, le fond, les boutons, et les cadres, ont été réalisées avec PhotoFiltre..

Il a été décidé de diviser le Poste Instructeur en deux parties distinctes : une partie dédiée à l'initialisation, qui sera effectuée avant chaque vol, et une partie dédiée à la gestion, pouvant être ajustée pendant le vol.

Dans la partie dédiée à l'initialisation, les actions comprennent la sélection des points de départ et l'initialisation des vérins. Cette étape, effectuée une seule fois au démarrage de la machine, permet de tester l'amplitude maximale et minimale de la plateforme afin de la calibrer.

Dans la partie dédiée à la gestion, les fonctionnalités incluent un bouton de gel permettant de mettre en pause le simulateur, ainsi qu'un bouton permettant d'activer ou de désactiver la plateforme. En outre, un bouton d'arrêt du simulateur est également disponible, totalisant ainsi 6 boutons au total.

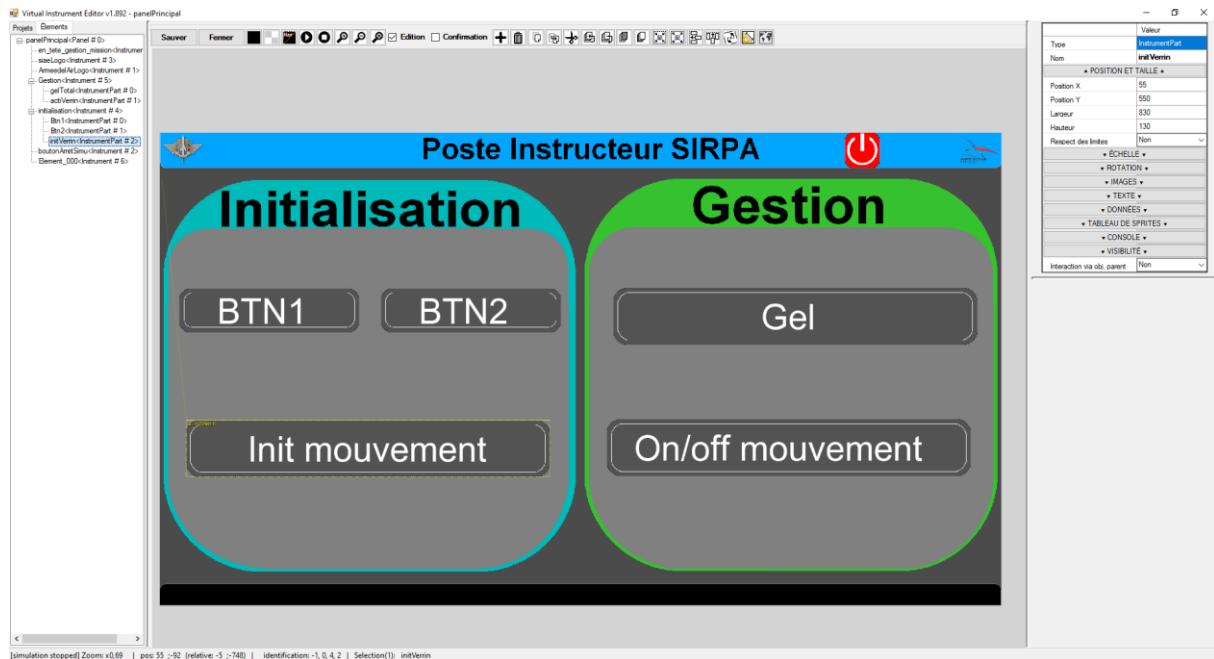


Figure 5. Virtual Instrument Designer et PI SIRPA

### 3. comprendre le programme précédent et l'adapter.

Pour éviter de recommencer à zéro, un accès au code précédent été disponible. La méthode consistait à conserver les éléments utiles, à supprimer ceux qui n'étaient pas nécessaires, et à s'inspirer des anciens boutons pour les ajouts, en observant leur conception.

Le programme est divisé en deux parties, la BDD et la partie onglet, chacune composée d'un .h et d'un .cpp. Le .h est la partie où sont déclarées les variables et fonctions, le .cpp est la partie où les fonctions sont codées.

Dans la BDD est stocké toutes les variables et fonction qui gardent une valeur en mémoire, l'onglet principal étant exécuté en boucle les variables se réinitialise constamment. Lorsque le programme de l'onglet utilise une variable de la BDD un pointeur est utilisé. Un pointeur pointe vers une fonction ce qui permet d'en récupérer uniquement ce qu'elle retourne.

Au lancement du programme, des fonctions d'initialisation sont appelées, incluant celles déjà présentes dans l'ancien programme ainsi que de nouvelles fonctions ajoutées selon les besoins.

Chaque bouton est programmé différemment car aucun ne fait appel à la même chose. Chaque bouton était une nouvelle découverte et une nouvelle façon de coder.

Les fonctions permettant de choisir un terrain appellent un fichier texte (.txt) extérieur au code dans lequel est écrit tout ce qui concerne la position de départ de l'avion, coordonnée, cap, roulis, tangage, etc etc. chaque point de départ est associé à un scénario qui peut également être activé via X-plane. Les scénarios ont été modifiés de sorte que l'avion soit toujours le SR20, excluant ainsi le choix entre le SR20 et le SR22. De plus, ont été intégrées au code, dans la fonction d'appui sur le bouton de terrain, les actions de rechargement d'essence, ainsi que d'initialisation de la température et de la pression atmosphérique. Par défaut en aviation l'atmosphère standard est définie par un QNH de 1013 et une température de 14.4°C au sol.

Le bouton gel est un bouton à 2 états, 2 images lui sont associés et chaque état de X-plane est associées à l'image mise en avant. Lorsque le logiciel est sur pause le bouton passe en vert et lorsque le simulateur est en marche le bouton est en gris. Pour ce faire le bouton est programmé de telle sorte à récupérer l'état de la dataref gel du logiciel X-plane, état qui sera stocké dans la BDD. Ainsi le bouton change d'état lorsqu'on appuie dessus ce qui met en pause le logiciel, et lorsque le logiciel est mis en pause manuellement l'état du bouton se met également à jour sur le PI.

Dans le programme qui a servi de base, aucune fonction ne permettait de gérer les mouvements. Il a donc été supposé qu'il était nécessaire d'utiliser des datarefs qui n'étaient pas encore disponibles, et dans lesquels il fallait écrire.

Dans un premier temps, un chemin a été créé dans le code jusqu'à un fichier pointant vers des datarefs non utilisés. Cette approche permettait de programmer temporairement toutes les fonctionnalités en attendant de recevoir les datarefs du fabricant. Une fois ces derniers fournis, il ne resterait plus qu'à modifier le nom du chemin dans le programme.

Finalement après avoir demandé au créateur de la plateforme, la société Brunner, nous a envoyé la doc technique sur les communications de la plateforme. Il s'est avéré que la plateforme ne communique pas à l'aide de datarefs mais des échanges de trames.

L'envoi de trame se fait par l'utilisation de ressources non présentes dans les fichiers principaux du programme. Il a fallu créer une DLL à part entière chargée de créer les trames et de les envoyer. À l'aide de la doc technique fournie les boutons ont été programmés de façon à envoyer les trames correspondant aux actions désirées.

Il y a 3 actions possibles, init, on et off, l'init ne se fait qu'une fois et ne peut être activé si la plateforme est en marche. On ne s'active que si l'init est terminé et que la plateforme est sur off, off se réalise lorsque la plateforme est active.

En plus d'avoir réussi à programmer les boutons comme désiré il s'est avéré que grâce à cette méthode il est possible de se passer du logiciel permettant de gérer la plateforme vérin. À l'origine la gestion de la plateforme vérin se faisait par un logiciel, cependant grâce à ces nouveaux boutons il est possible de gérer les vérins sans passer par le logiciel.

Un effort a été fait pour organiser le code de manière claire et ajouter des commentaires expliquant les actions effectuées. Cela permettra aux futurs intervenants de comprendre le code plus facilement.

#### 4. débogage

L'étape la plus longue de la programmation est le débogage, elle représente  $\frac{3}{4}$  du temps qui est passé sur un programme, comprendre l'erreur et la « réparer ».

Pour se faire il existe différentes méthodes selon l'erreur rencontrée. Il peut s'agir de simples erreurs de code, mauvais appels de fonctions, variables inutiles etc...

Il existe plusieurs types d'erreurs qui ont été fréquemment rencontrées lors de la programmation du PI. Les erreurs de "link" se produisent lorsque des symboles ne sont pas définis, lorsqu'il y a plusieurs

symboles définis, ou encore lorsqu'il y a des erreurs de librairie ou des conflits de noms. Dans ces situations, il suffit de localiser le pointeur problématique et de remonter la fonction qu'il appelle.

Lorsque le programme fonctionne mais ne réalise pas ce qu'il est souhaité ou ne fonctionne pas correctement, il est nécessaire de trouver l'origine de l'erreur. Pour ce faire, des "points d'arrêt" sont utilisés : le programme démarre normalement puis se met en pause à la ligne de code où le point d'arrêt a été placé. Ensuite, le programme est exécuté ligne par ligne à partir du point d'arrêt jusqu'à ce que la source de l'erreur soit identifiée.

Enfin, le problème le plus intéressant à résoudre a été celui rencontré lors des premiers essais du programme sur la partie vérin. Initialement, le programme ne fonctionnait pas correctement, le programme n'utilisant pas de datarefs mais des trames qu'il fallait intercepter pour vérifier l'envoi du bon message.

Pour résoudre ce problème, un ordinateur portable équipé du logiciel Wireshark a été utilisé. Il a été connecté au réseau d'échange à l'aide d'un hub. Initialement, le test avait été réalisé avec un switch, mais la lecture des communications s'est avérée impossible. Une fois le hub connecté et après avoir vérifié que toutes les communications étaient interceptées, des filtres ont été appliqués pour n'afficher que l'échange qui nous intéressait.

La machine sur laquelle le PI est installée a pour adresse 192.168.0.1 et la plateforme vérin a pour adresse IP 192.168.0.20, les filtres WireShark ont donc été appliqué de façon à n'avoir que l'échange de trames entre le PI et la plateforme.

En figure 6 il est observable que la trame indiquée en « data » ne correspond pas du tout à celle qui est censé être envoyé (annexe 6). À partir de cette information il a été compris que le problème venait de la partie du programme qui créait les trames.



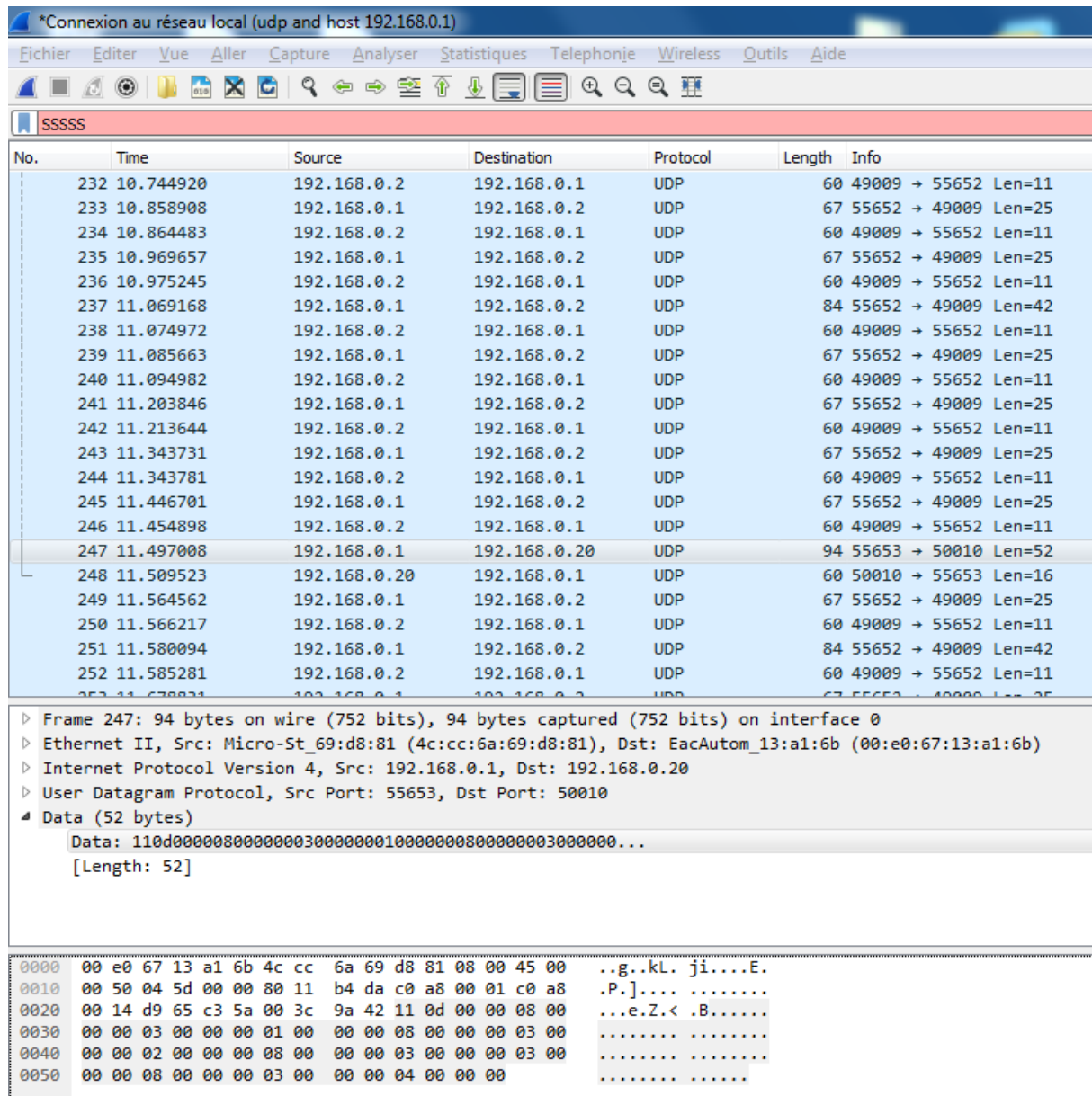


Figure 6. Échange de trame intercepté entre le PI SIRPA et la plateforme vérin

5. Le simulateur qui doit être livré peut se diviser en 3 parties, le PI, dont la réalisation m'a été confié, le simulateur avec tous les appareils de mesure, les écrans, l'électronique et la plateforme à vérins, et enfin les pc sur lesquels sera stockés l'entièreté des logiciels et programme.

Dans le bureau les pc utilisés sont des desktop, des ordinateurs « classiques », cependant lorsque le simulateur est livré il faut que tous les pcs soient fournis en 1 bloque afin facilité le transport,

l'utilisation et la gestion de place. Pour cela il faut utiliser des baies de serveur, une baie de serveur est composée d'une armoire informatique et de pc industriels qui seront vissés à l'armoire. La nouvelle tâche qui m'a été confié après avoir fini la précédente est de trouver des armoires informatiques pouvant contenir 5 pc industriels de 4 u de haut chacun. Les u sont les unités de mesures de rack informatique, 1 u = 4.445 cm. En plus des 5 pc de la place supplémentaire est à prévoir puisqu'un kvm switch sera installé, un appareil permettant de changer rapidement quel pc est projeté à l'écran, une multiprise et autres matériels.

Sachant que les PC feront 4 U de hauteur, les recherches ont commencé par l'armoire informatique. Les critères de recherche comprennent une baie informatique d'au moins 20 U de hauteur (taille standardisée d'armoire par des multiples de 3), de 19 pouces de large (largeur des PC industriels du simulateur récupérés lors de la mission à Salon), et un critère très important : lors de la mission à Salon-de-Provence, il a été nécessaire de déplacer deux baies informatiques qui étaient sur des pieds fixes, et ces baies se sont avérées extrêmement lourdes pour leur taille. Par conséquent, il a été décrété que l'un des critères principaux lors de la recherche serait de choisir une armoire équipée de roulettes freinables.

Après analyse des PC chargés de faire tourner le simulateur présent dans le bureau, une configuration semblable à celle-ci a été envisagée :

- pc rack 4U
- cpu i7 ou équivalent
- ram 16 Go minimum
- stockage 1To
- NVidia GTX 1080 ou équivalent
- alimentation standard
- os windows

### 2.3.2. Mission secondaire

La base militaire de Salon-de-Provence a sollicité nos services afin de se débarrasser d'un simulateur. Une mission militaire d'une durée de 2 jours a donc été organisée, la mission consistait en le démantèlement d'un simulateur à l'école militaire de Salon-de-Provence.

L'objectif de ma participation à cette mission était, en plus d'aider, de voir l'usage qu'il est fait du simulateur sur lequel je travaille au bureau. Le simulateur a été démantelé, chargé dans un camion puis ramené à la base de l'AIA de Cuers comme nous le demandait la mission.

Profitant d'être sur place, une inspection a été menée sur les autres machines en service et des problèmes ont été constatés, certains appareils de mesure sont endommagés, et sur certains simulateurs fonctionnant encore sur d'anciennes versions, les nouveaux palonniers ne sont pas compatibles. Cependant par manque de temps il nous était impossible de réaliser cette tâche, il est donc nécessaire de repartir en mission au plus vite afin de régler ce problème.

Une maintenance sur le système électronique devra être réalisée. Malheureusement, il ne m'a pas été possible de participer à cette seconde mission.

## 3. Bilan d'activité

### 3.1. Présentation des résultats

Avant mon arrivée le projet SIRPA n'avait pas encore démarré, cependant le projet Cirrus lui était terminé. Le simulateur d'entraînement de l'armée était déjà opérationnel et en fonction sur base, il ne restait qu'à adapter le simulateur au nouveau projet. Le cahier des charges donné par le SIRPA manquant d'énormément d'information il m'a été possible de faire avancer le projet comme je l'entendais. À la suite de mon stage la partie software du projet est terminée ou presque ce qui représente une grande partie du projet. Une partie des recherches de matériels destinés à stocker le programme et le logiciel du PI SIRPA a également été réalisée.

### 3.2. Prolongement de mission possible

La mission que l'on m'a confiée s'arrête à la réalisation du PI, cependant le projet total n'est pas terminé, il reste la partie installation matérielle des baies informatiques ainsi que la modification simulatrice. En effet le simulateur présent dans le bureau est adapté au CIRRUS mais pas au SIRPA. Le tableau de bord doit être aménagé d'un écran tactile en face du siège copilote, le modèle 3D SolidWorks devra d'abord être modifié, les pièces seront ensuite imprimées en 3D et implémentées au simulateur dans le bureau afin de réaliser des tests.

Un prolongement de mission possible serait de refaire toutes les mesures du simulateur présent dans la pièce, vérifier que les mesures du modèle 3D sont conformes au simulateur réel et modifier le modèle 3D afin d'y ajouter un écran tactile à l'emplacement requis conformément au cahier des charges.

### 3.3. Apports personnels

Ce stage a été très instructif sur de nombreux points que cela soit en rapport avec mon projet ou non. Il m'a permis d'en apprendre plus sur l'assemblage et la maintenance des aéronefs et hélicoptères que cela soit au niveau mécanique ou au niveau électronique.

Des connaissances supplémentaires ont été acquises en aviation et dans le domaine militaire, tout en progressant dans la programmation en C++.

Cette expérience m'a également permis d'énormément m'améliorer en programmation que cela soit l'organisation, la compréhension, mon niveau en général ou la résolution de problèmes. Pour surmonter certaines difficultés, il m'a été possible de bénéficier de l'aide de mon tuteur ainsi que d'autres employés lorsqu'il n'était pas disponible. Cela m'a appris à ne pas hésiter à demander de l'aide en cas de besoin, sachant qu'il y a toujours quelqu'un pour m'aider ou me guider. Ce qui mène également à une autre leçon apprise, il est important de s'entendre avec tout le monde et tisser des relations afin d'avoir le plus de main possible en cas de problèmes.

## Conclusion

Le stage était axé autour d'une mission unique qui a pris plusieurs semaines à réaliser, impliquant principalement de la programmation ainsi qu'un peu de transport de matériel.

La problématique de ce stage était « comment mener à bien l'adaptation d'un simulateur d'entraînement de l'armée en une « attraction » grand publique ? », l'adaptation s'est faite en simplifiant au maximum : tous les éléments paramétrables dans le simulateur militaire sont devenus des paramètres définis à une valeur par défaut et l'interface a été conçue et optimisée pour une utilisation simplifiée.

La complexité du code, étant d'un niveau supérieur à celui abordé en cours, ce qui m'a permis de parfaire mes acquis scolaires. J'ai pu approfondir mes connaissances et développer de nouvelles compétences. Au fur et à mesure, la compréhension du code s'est améliorée, ce qui m'a permis de progresser efficacement dans mes tâches. La saé concevoir robotique m'a beaucoup aidé lors de l'étape de compréhension de structure du code et utilisation de logiciel. Les compétences acquises lors de cette saé m'ont permis d'avoir des bases solides.

Les cours de réseau ont également été très utiles lors de l'analyse des trames. Grâce à ces cours, j'avais déjà appris à utiliser le logiciel WireShark, à lire les communications et à les filtrer.

Ce stage m'a permis de confirmer mon envie de travailler dans ce secteur, rien ne m'a déçu durant ce stage, chaque jour je venais au travail avec le sourire et je n'avais pas l'impression de travailler mais de m'amuser tout en aidant l'entreprise dans ses projets.

## Glossaire et liste des figures

### Glossaire

AIA-CP : Atelier Industriels de l'Aéronautique de Cuers-Pierrefeu

SIAé : Service Industriel de l'Aéronautique

IHM : Interface Homme - Machine

PI : Poste Instructeur

PI Cirrus : Poste instructeur conçu pour l'école militaire

Cirrus SR 20/22 : Avions américains construits par Cirrus et utilisés dans l'armée de l'air

PI SIRPA : Poste instructeur demandé par le SIRPA que je dois réaliser

SIRPA : Service d'Informations et de Relations Publiques des Armées, le service de communications des armées françaises

X-plane : Un logiciel de simulation de vol développé par Laminar Research. Il s'agit d'un simulateur amateur ou professionnel, X-Plane propose des avions commerciaux ou militaires.

Virtual Instrument Designer : logiciel permettant de réaliser l'IHM

Photo Filtre : un genre de paint plus poussé, nul à chier mais j'ai dû faire les éléments de mon IHM dessus

SIOC : Un logiciel grand public qui permet de connecter des cartes électroniques pour récupérer l'état des boutons. Il possède un script qui marche grâce aux événements reçus.

Dataref : c'est un concept utilisé principalement dans le domaine des simulateurs de vol, notamment dans X-Plane. Elles permettent de lire et écrire des données spécifiques au simulateur de vol en temps réel. Elles permettent de connaître en temps réel divers paramètres comme la position, la vitesse, paramètres d'instruments etc... Toutes les informations propres au simulateur sont gérées par des datarefs. Contrôler les datarefs revient à contrôler le simulateur.

BDD : Base De Donnée d'un programme

## Liste des figures

Figure 1. Hiérarchie des structures dont dépend l'AIA-CP

Figure 2. Vue aérienne de l'AIA-CP

Figure 3. Organigramme du pôle conception

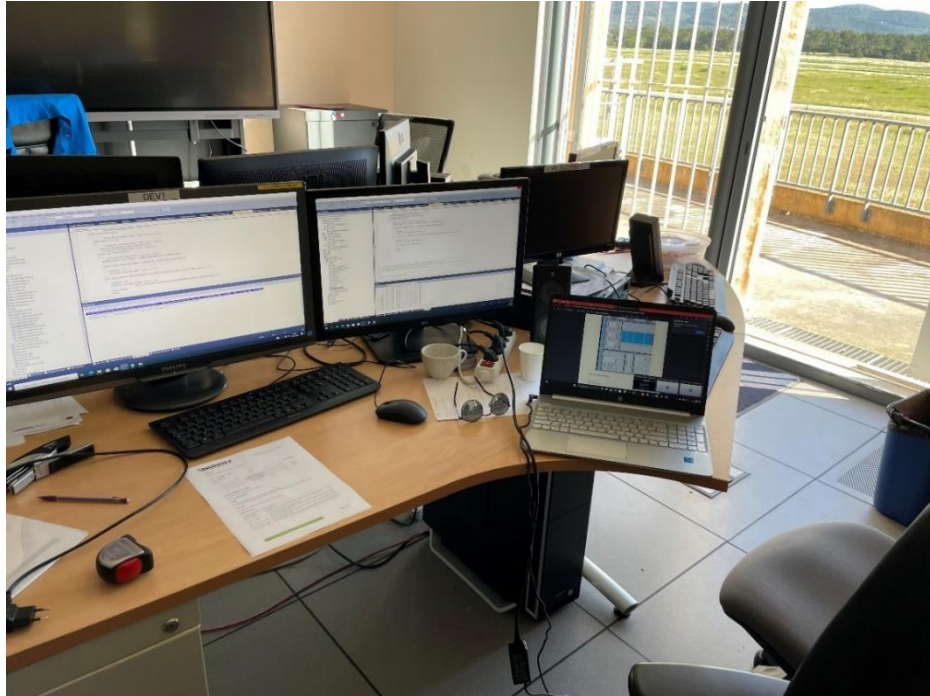
Figure 4. Poste instructeur Cirrus

Figure 5. Virtual Instrument Designer et PI SIRPA

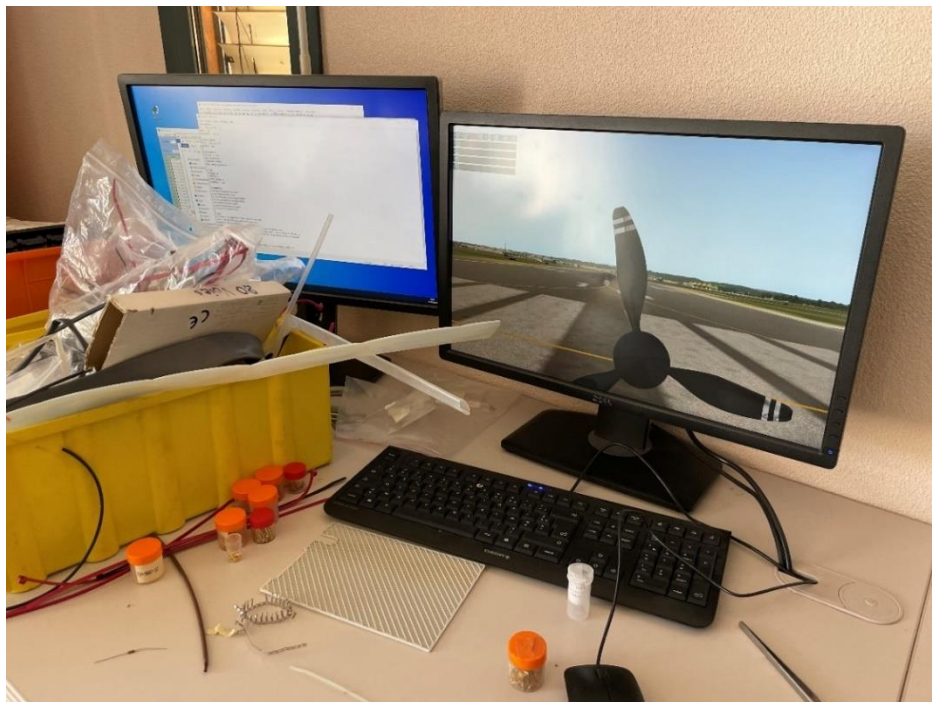
Figure 6. Échange de trame intercepté entre le PI SIRPA et la plateforme vérin



## Annexes

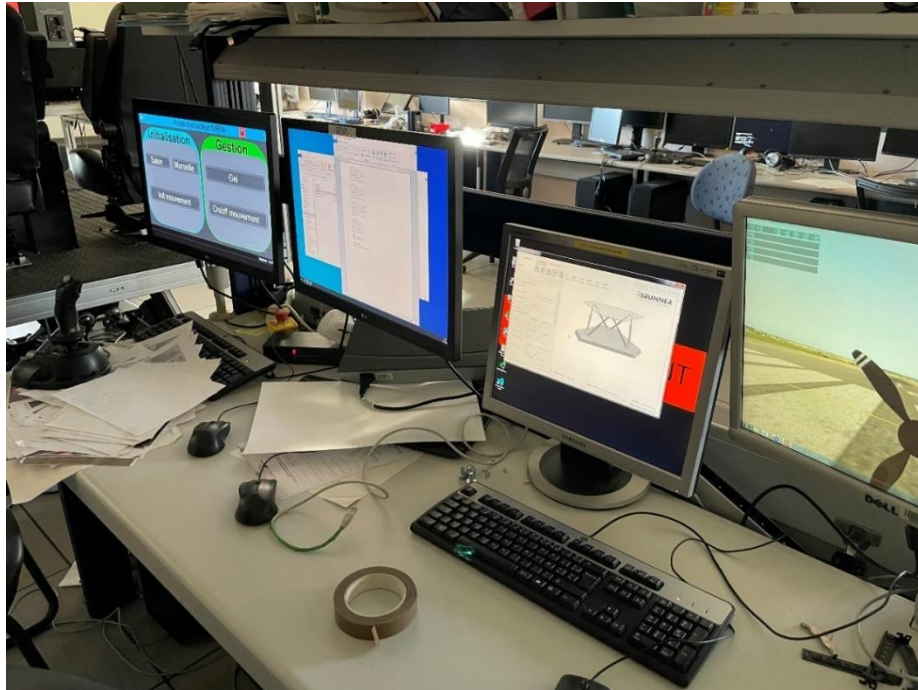


Annexe 1. Poste de travail principal



Annexe 2. Simulateur de test

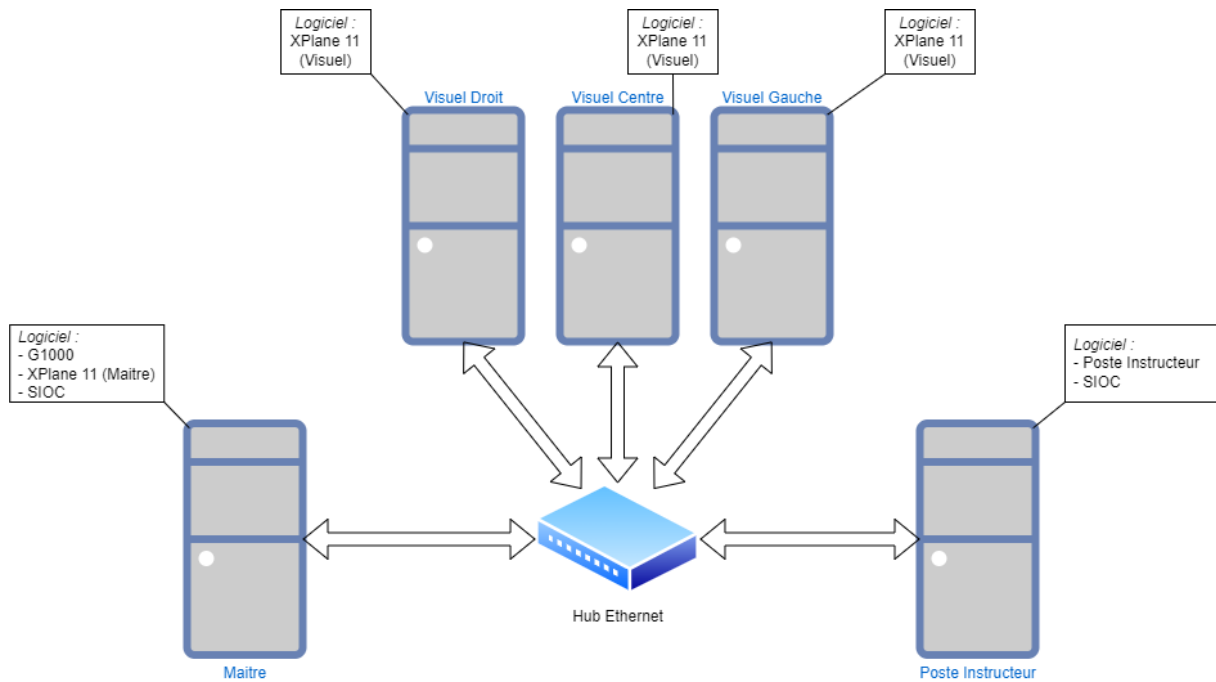




Annexe 3. Poste de contrôle du simulateur (PI, X-plane, contrôle vérin)



Annexe 4. Simulateur avec plateforme à vérins



Annexe 5. Explication de la communication des machines

## Min latency

Min drive communication latency in microseconds.

### 2.3.4 Transition to state

#### 2.3.4.1 Request

Offset	1	2	3	4	Field datatype
0	Length (0x00000008)				UInt32
4	Command (0x00000003)				UInt32
8	Requested State				UInt32

#### Example recorded from traffic

08 00 00 00 03 00 00 00 01 00 00 00

#### 2.3.4.2 Request Fields

##### Length

Contains the byte length of the request, minus this field.

For this packet, the packet size is fixed at 12 bytes, so the length field will contain the value 8.

##### Command

Contains the command that was sent. 0x00000003

##### Requested State

The state the platform should transition to.

Value	State	Description
1	Stop	Halts all drives at their current position. Stops motors and engages brakes. Platform will need to be initialized again.
2	Init	Searches the lower endpoints of all drives, performs a weight measurement and moves to parking position.
3	Parking Pos	From active state (either sim or manual mode) move to parking position. Stop motors, engage brakes.
4	Sim Mode	Move to center position, then enter simulation mode. Listen for SimData packets.